

# Towards Automated Integrity Constraints Modelling and Validation

## *A Survey and Approach*

Sandra Bergner

*Institute for Applied Computer Science, Universität der Bundeswehr,  
Werner-Heisenberg-Weg 39, 85577, Neubiberg, Germany*

Keywords: Knowledge Engineering, Semantic Web.

Abstract: Semantic web techniques based on ontologies are a possible means for modelling and validating complex, safety-critical products like airplanes or automobiles. For validation purposes, checks based on the Open World Assumption (OWA) as well as checks based on the Closed World Assumption (CWA) are both valuable. Based on a survey of existing semantic-based approaches, we present a novel approach that provides hybrid OWA/CWA checks and thereby reduces the maintenance burden for managing two different kinds of checks.

## 1 INTRODUCTION

The design of complex, safety-critical products like airplanes or automobiles is normally performed by distributed, concurrent engineering teams. During development, the overall system specification emerges from several fragments of different sources. Airbus, for example, is using ontologies based on the OWL 2 ontology modelling language to specify the product models of their aircrafts (Bergner, 2015). Several domain engineers integrate their knowledge into a common model of the aircraft. This common model covers concepts from domains like cabin layout, lighting and electrical wiring.

Airbus checks the consistency (Hitzler et al., 2007) of these domain models by using OWA checks performed by existing reasoners like Pellet (Sirin et al., 2007).

Based on these consistent models, airline customers like Lufthansa, Air Berlin or Emirates add information to configure specific aircraft exemplars. To check the completeness of such a specified aircraft configuration, they have to switch from the OWA to the CWA.

A real-world example for this transition arises, for example, in the electrical wiring domain to ensure that the connectors and sockets are correctly plugged into each other. On a first view, it seems rather simple to guarantee that, but imagine the great number of cables that are necessary in an airplane (see both sides of Figure 1).

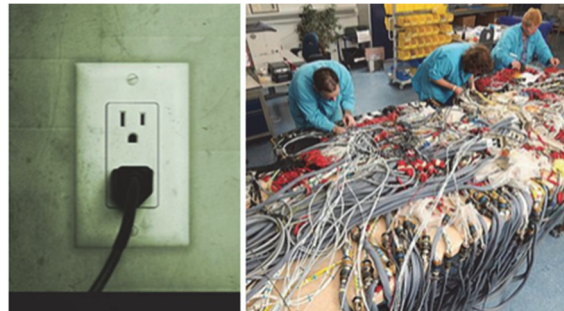


Figure 1: Connectors Plugged into Sockets.

In the electrical wiring scenario, OWA checks are well suited to detect inconsistencies like connectors that are plugged into the wrong sockets. However, OWA checks cannot easily detect whether the wiring configuration is complete, that is, whether there is a corresponding socket for each connector, and whether all connectors are actually properly connected and don't dangle around.

Unfortunately, appropriate methods and tools that combine OWA and CWA reasoning for system validation do not exist yet. In the database and knowledge management communities, CWA checks are also known as integrity constraints (ICs). There are many approaches on the topic of ICs, so that getting an overview of the state of the art is a challenge in itself. Thus, one of the main contributions of this paper is to introduce and review the different fields of research within the last couple of years. To do this, Section 2 reviews the existing

work with regard to the following research question: Is there a practical approach that enables an automated transformation of OWL 2 axioms to ICs, thereby bridging the OWA/CWA gap?

Based on that survey, Section 3 presents a novel approach for hybrid OWA/CWA checks based on a combination of some of the most advanced existing approaches. Section 4 then shows how these checks can be automated based on existing tools and libraries. Finally, Section 5 gives a short conclusion and provides an outlook on further work.

## 2 EXISTING APPROACHES

As motivated in the introduction, the derivation and handling of ICs for completeness checking is the focus of this paper. Thus, we first survey and evaluate the existing semantic-based approaches. Based on that evaluation, we identify a set of “compatible” approaches and techniques that can be combined to achieve hybrid OWA/CWA checks.

All of the surveyed approaches are semantic-based, that is, they use the same syntax with two different semantics, one for the OWA and one for the CWA.

### 2.1 Auto-Epistemic Extensions

Research on Integrity Constraints in the field of auto-epistemic extensions is based on Reiter’s statement (Reiter, 1992) that “constraints are epistemic in nature; rather than being statements about the world, they are statements about what the knowledge base can be said to know.”

Based on that, Reiter points out that ICs should be understood as epistemic First Order Logic (FOL) sentences. More generally, his approach is based on static constraints and the assumption that both knowledge bases and ICs are a set of first order sentences.

For example, the fact that every resident of the USA must have a social security number is written as a first-order logic sentence as follows:

$$(\forall x)resident(x) \supset (\exists y)ss\#(x, y) \quad (1)$$

Reiter interprets this statement as concerning the contents of the knowledge base. Thus, he introduces the modal K operator (for “known”), resulting in the following first-order logic sentence:

$$(\forall x)Kresident(x) \supset (\exists y)Kss\#(x, y) \quad (2)$$

Hustadt agrees with Reiter’s view that an epistemic K operator should be used for handling closed world

assumptions (CWA) in knowledge representation languages (Hustadt 1994). Hustadt underlines the non-decidability of theorems with K operators. Furthermore, he suggests to “identify a fragment of  $NLL$  to which we can add an epistemic operator without losing decidability” (Hustadt 1994).

One step in this direction is the approach from Donini et al. in (Donini et al., 2002), which is based on the epistemic extension of Description Logics (DLs). They suggest a Framework of Description Logics with minimal knowledge and negation as failure (MKNF-DLs).

To do this, they first divide the knowledge base into two components called the „TBox“ (containing the concepts and roles) and the „ABox“ (containing the individuals). The framework then augments DLs with modal operators K and A both on roles and concept expressions. These operators are interpreted according to the non-monotonic logic MKNF (Lifschitz 1994). Hence, KC is defined as the set of individuals that are known to be instances of concept C, and KR is defined as the set of pairs of individuals that are known to be instances of the role R. Furthermore, the A operator is introduced as an auto-epistemic assumption.

Donini’s approach is based on Reiter’s view (Reiter 1992) that ICs should be usually regarded as queries. This results in the epistemic description logic ALCK (Donini et al. 1992). Thus, ICs are represented as epistemic DL axioms. The standard DL KB defines the satisfaction of ICs as the entailment of the epistemic IC axioms.

In (Grimm and Motik 2005), Motik et al. propose an epistemic operator to realize a non-monotonic extension of OWL. The approach is based on auto-epistemic description logics (ADL) (Donini et al. 2002). “In other words, we believe that many applications require OWA and CWA in parallel in order to enable local closed world (LCW) reasoning” (Grimm and Motik 2005). For them, such reasoning is based on “the OWA augmented by the possibility to explicitly close off parts of the world.” (Grimm and Motik 2005). They introduce three methods for local closed world reasoning based on epistemic operators.

The approach in (Katz and Parsia 2005) by Katz and Parsia is also based on Reiter’s view on integrity constraints, namely “asking the data only for known facts”. They discuss “ALCK, a non-monotonic logic that augments ALC with the epistemic operator K and argue that a similar extension to OWL is desirable”. The authors point out that the OWL syntax is too inflexible to be extended in a natural way. Therefore, they introduce their implementation

of ALCK as an extension to the tableau-based OWL-DL reasoner Pellet (Sirin et al., 2007).

Another direction is to combine the Semantic Web Rule Language (SWRL) and DL-safe rules with the approaches for auto-epistemic extensions of DLs. To do this, Motik et al. propose a novel logic for hybrid MKNF knowledge bases similar to (Donini et al., 2002), which is based on Lifschitz (Lifschitz 1991). In addition, their “logic seamlessly integrates OWL with Logic Programming” (presented in (Motik et al., 2006)). Finding well-founded semantics, “which is nontrivial because it actually requires redefining the semantics of MKNF” (Motik and Rosati, 2006) remains an open point.

An epistemic query based approach is presented in (Calvanese et al., 2007). This approach is based on expressing ICs as epistemic queries. It evaluates ICs by checking the epistemic query answer. The epistemic queries are based on EQL-Lite (Q), which is a fragment of EQL, a first order modal language with equality and a single modal operator K.

The approach from Tao et al. (Tao 2010) reduces Integrity Constraint validation to SPARQL query answering and is based on Reiter’s view that “IC should be epistemic FOL queries that will be asked to a standard KB that does not contain epistemic axioms.”

Based on (Tao, 2010), Tao suggests an IC Semantics for OWL 2 that is different from the standard OWL 2 semantics, but keeps the full expressivity of SROIC DL. Furthermore, she proposes a concept for IC validation that includes the translation of rules from ICs to DCQ<sup>not</sup> queries. Her “translation rules are similar in the spirit to the Lloyd-Topor transformation” (Lloyd, 1987). However, instead of generating rules, her core idea is to translate IC axioms into negated queries, so-called DCQ<sup>not</sup> queries. In case the IC is violated, the KB entails the query. That means, if the answer of the query is not empty, one can conclude that the IC is violated.

On this basis, Tao et al. have built the prototype IC validator Stardog (Clark and Sirin 2013) as an extension of the OWL 2 DL reasoner Pellet (Sirin et al. 2007). Their prototype reads ICs expressed as OWL axioms and then translates each IC first into a DCQ<sup>not</sup> query and then into a SPARQL query.

## Evaluation

With regard to auto-epistemic extensions, the approach of Tao et al. is the most advanced. The approach has proven its feasibility and provides the full expressivity of the SROIC DL. Furthermore, it

has been integrated into the Stardog tool (Clark and Sirin, 2013), whose OWL 2 DL reasoner has been enhanced by closed world reasoning. Although the approach is already used in practice, we see it as a significant disadvantage that the user still needs to have knowledge of the Integrity Constraints and needs to know how to formalize these constraints. This results in significant modeling burden.

## 2.2 Circumscription

The main purpose of Circumscription approaches is to not consider the whole knowledge, but only the knowledge that is designated as closed world information.

In (Etzioni et al., 1994), Etzioni et al. describe a method for representing, inferring and updating local closed world (LCW) information. They address the problem of “redundant information gathering” in (partial-order) planning. They state that “The agent is not making a closed world assumption. Rather, the agent has access to an action that yields closed world information.” Consequently, they start with the notion of an incomplete world. They formalize the agent’s incomplete information by a set of possible world states,  $S$ . Those states are consistent with its information. As they assume that the agent does have correct information, the current world state  $w$  is necessarily a member of  $S$ . Accordingly,  $S \models \varphi$  means that  $\varphi$  is known by the agent, just in case  $\forall s \in S, s \models \varphi$  applies. They assume that the agent possesses complete information if  $S$  and  $w$  entail exactly the same facts. Consequently, incomplete information denotes that there are facts such that neither  $S \models \varphi$  nor  $S \models \neg \varphi$ ; this means that  $\varphi$  is unknown to the agent.

Hence, Etzioni et al. maintain a meta-level database DC that contains sentences of the form  $L(\Phi)$ , which record explicitly where (in the database) the agent has closed world information. Furthermore, they provide inference rules for LCW as well as update rules for maintaining the consistency of LCW with the world state.

“Agents must plan how they can achieve their goals” (Heflin and Muñoz-Avila, 2002), particularly with regard to the available information for that they cannot make a closed world assumption. Via using Local Closed world (LCW) information (Heflin and Muñoz-Avila, 2002), they approach this problem by explicitly stating which information can be assumed to be complete. The approach is based on extending two semantic web languages with the ability to state LCW information. “LCW information is given as

meta-level sentences of the form  $LCW(\Phi)$ . The semantics of  $L(\Phi)$  is that for all variable substitutions  $\theta$ , if the ground sentence  $\Phi\theta$  is true in the world, then  $\Phi\theta$  is represented in the agent's knowledge base." (Heflin and Muñoz-Avila, 2002). Consequently, if a matching ground sentence is not in the knowledge base, it is known to be false.

Furthermore, they suggest a concept that enables agents to plan efficiently in distributed information environments. The concept is based on ordered task decomposition (OTD) in combination with LCW information. It is an extension of OTD (Dix et al. 2003) that takes advantage of local closed world information in the context of the semantic web. A key component of the approach is the Mediator, whose "main function is to evaluate the OTD Plan Generator's preconditions by accessing remote information sites." (Dix et al., 2003). One kind of the LCW information is provided explicitly by the information sources and is stated as locally closed.

The approach from Krishnathi et al. in (Krishnathi et al., 2011) is a combination of open and closed world reasoning – also called Local Closed World Reasoning. The approach is based on "Grounded Circumscription" and adapts the circumscription description logic, which is applicable to *SROIQ*. It uses a knowledge base that is based on description logic as usual. The authors designate some predicates (concept names or role names) as closed via augmenting them with meta-information. More precisely, they "simplify the circumscription approach by restricting their attention to models in which the extension of the minimized predicates may only contain known individuals from the KB" (Krishnathi et al. 2011). They have pointed out that it would be preferable to obtain a language, which is intuitively very simple to understand by ontology engineers.

### Evaluation

In the context of Circumscription, Etzioni et al. have presented a sound and computationally tractable method for representing, inferring and updating Local Closed World Reasoning. The approach from Krishnathi et al. is applicable to *SROIQ*, but they still have to investigate the complexity of their approach. Both approaches are integrated into prototypes but are not proven in practice. Thus, as of now they do not provide a language for the closed world reasoning which is intuitively very simple, appeals to ontology engineers and is computationally effective.

## 2.3 Rule based Formalisms

As introduced before, ICs need some sort of CWA. "One way to add CWA to OWL is via integration with logic programming (LP)." (Sirin et al. 2008). In this line of approaches, "LP provides negation as failure under CWA and thus can be used to express ICs." (Sirin et al., 2008).

The main idea here is to extend DLs by first-order rules. In (Lloyd and Topor, 1984), Lloyd et al. are using first-order formulas to express queries and integrity constraints based on PROLOG. The "requirement of implementing such a feature is a sound form of the negation as failure rule." (Lloyd and Topor, 1984).

Mäs et al., (Mäs et al., 2005) introduce a framework for using ontologies for the definition of ICs. Those constraints are formalized in SWRL (Semantic Web Rule Language), which is a combination of OWL and RuleML (Rule Markup Language). Thus, Mäs et al. are using integrity constraints within ontologies that are expressed through axioms.

The approaches in (Damásio et al. 2006)(Polleres et al. 2006), (Lukasiewicz 2004) are based on logic program transformation. Motik et al. (Motik et al. 2006) combine the logic program transformation with auto-epistemic extensions. The approach in (Motik et al., 2006), (Lukasiewicz ,2004) achieves the integration of ICs with OWL by using a rule-based formalism. The approaches are based on Hybrid KBs (DL and KB).

The goal of (Polleres et al., 2006) by Polleres et al. is to provide a context-aware rule language. The core idea is to view scoped negation as an extension of RDFS. Their outcome is a basis for a rule based query language and they suggest SPARQL or N38 as an appropriate candidate to extend negation-free RDF.

Schmidt et al. (Schmidt and Lausen, 2013) propose a concept of how to consume linked data with RDF Data Descriptions (RDDs). They present design goals, syntax and formal semantics for RDDs based on first-order logic. Constraints are defined by first-order sentences known as tuple-generating and equality-generating dependencies, which can be implemented as SPARQL ASK queries. This is due to the proof "that every first-order sentence can be expressed in SPARQL" (Schmidt and Lausen, 2013).

### Evaluation

This research area is still in its infancy. We disagree with the opinion of Mäs et al. that constraints should

be part of the ontology and should not be treated separately. We think that integrity constraints should be treated separately from the ontology, because they do not constitute knowledge about the world, but are instead a way to check this knowledge. In our opinion, it is important that the original state of an ontology is retained, because we will use this ontology or parts of it as a base model for specifying a number of different products, each with specific properties. This requires that the consistency and completeness checks can be performed for each of the derived products separately. Thence, in our opinion the models (TBox), the products (ABox) and the constraints (CBox) have to be orthogonal in order to be able to combine and reuse them arbitrarily.

Motik et al. extend the logic program transformation by the auto-epistemic  $K$  operator. However, following this approach, ontology developers have to deal with two worlds. On the one hand, they have to state rules, and on the other hand, they have to model the domain with the ontology language OWL. This results in considerable modeling burden.

The approaches from Polleres and Schmidt both suggest SPARQL as an appropriate candidate for formalizing integrity constraints, but their approaches are still on a conceptual level.

## 2.4 Special Purpose Boxes

The semantics of special purpose box approaches is based on the notion of outer skolemization of first-order logic formulae and minimal-model semantics. The key idea in (Motik et al., 2007) Motik et al. is to apply the approach of ICs from relational databases to OWL. The approach introduces “extended DL knowledge bases, which allow a modeller to designate a subset of the TBox axioms as ICs” (Motik et al. 2007). For ABox reasoning, these axioms are interpreted as checks, with the intention to check the proper form of the ABox. “Introducing distinct individuals for each existential quantifier can be justified by skolemization, the well-known process of representing existential quantifiers with new function symbols” (Motik et al., 2007).

Another approach in this line from Motik et al. (Motik et al., 2009) is based on minimal Herbrand models. The core idea is to “augment OWL with ICs” (Motik et al., 2009). An “OWL IC axiom is satisfied if all minimal Herbrand models of the KB satisfy it” (Motik et al., 2009). By using this approach, one only needs to tag some TBox axioms as being ICs. Subsequently, those axioms cannot

imply new ABox facts, but are only used to check whether an ABox axiom is in the appropriate format. In contrast, “a constraint axiom can still imply a new TBox axiom, e.g., a subclass relation might be inferred by using ICs. (Sirin et al., 2008)”

Seylan et al. propose a DBox approach (Seylan et al. 2009) that leverages the standard semantics of relational databases. A DBox includes atomic class or property assertions axioms such as “penguins are birds”. Furthermore, a DBox includes the extensions of predicates (classes and properties), which are bounded by the DBox. Thus, predicates that do not appear in the DBox remain open. Correspondingly, the setup “generalizes both standard OBDA (only open predicates permitted) and DBoxes (only closed predicates permitted in data)” (Seylan et al., 2009) and integrates open and closed predicates in one ontology.

The core idea described in (Patel-Schneider and Franconi, 2012) by Patel-Schneider et al. is to “directly state that the extension of certain concepts and roles is complete by making them DBox predicates.” This proposal might eliminate the need for special semantics.

In contrast to “the DBox approaches, the NBox approach supports deduction on closed concepts and roles” (Ren et al., 2010). Ren et al. describe an approach “extending the syntax of DL SROIQ with an NBox” (Ren et al., 2010). The NBox extends the semantics with the concept of Negation as Failure in order to specify the predicates to close. This procedure is implemented in the TrOWL9 infrastructure. In (Pan and Ren, 2012) Pan et al. allude to “the fact that NBox LCWR requires support to enumerations (sets) in the ontology” (Pan and Ren, 2012). It is still an open issue how enumerations should be realized in a language that does not support enumerations.

In (Lutz et al., 2013) Lutz et al. “carry out a non-uniform analysis of the data complexity for query answering with closed predicates”. “Admitting closed predicates in OBDA” results in the problem “that query answering becomes intractable regarding data complexity” (Lutz et al., 2013).

### Evaluation

The approach from Motik et al. is still on a conceptual level. We agree with their idea that a certain subset of TBox axioms can be designated as constraints, but the question of how to designate such axioms as constraints must still be solved. Furthermore, the approach has to be transferred to practice.

The DBox approach from Seylan et al. is a good basis for our approach because it separates the ABox, the TBox and the DBox, whereupon the DBox should only include closed predicates. Hence, those predicates that are not included in the DBox remain open. The limitation of this approach is that inference based on DBox predicates is prohibited in a sense that no new instance can be inferred. Furthermore, the disadvantage of this approach is that it considers only the basic, propositionally closed DL *ALL* and not more expressive DLs like *SROIQ*.

We agree with the idea described by Patel-Schneider et al. that DBox should be made into the analogue of database tables to eliminate the need for special semantics.

### 2.5 Evaluation with Respect to the Research Topic

The previous sections about the related work show that the state of the art is very heterogeneous, typical for the early stage of a technology that is not quite ripe for practical applications. As could be seen, there are many competing approaches and various tools by different research teams and companies. The challenge is to bring the theoretical approaches into practical application by defining an approach with the following properties:

- **Formally Sound:** The approach combines existing, well-founded methods and techniques for knowledge representation and for consistency and completeness checking with regard to OWA and CWA.
- **Usable in Practice:** The approach hides the mathematical complexity of the formal methods and techniques from the user, automating the consistency and completeness checks.

In order to achieve that, we select techniques that are formally sound and sufficiently expressive to support all necessary use cases for consistency and completeness checks of complex, real-world knowledge bases. Thus, we have chosen the following techniques for our approach:

As the basis of our approach lies the auto-epistemic extension technique from Tao (Tao et al. 2010) (see 2.1.1 Auto-epistemic Extensions). This approach is formally sound and offers the full expressivity of SROIQ. It already supports both open world and closed world semantics for single OWL 2 axioms as well as for complex expressions.

Furthermore, it is already well tested in real-world scenarios.

As currently implemented, Taos approach is rather laborious in practical applications, as the user has to manually enter the same axioms twice, once for the OWA checks and once for the CWA checks. To improve that, we use syntactic axiom annotations (Motik et al. 2009) that denote whether an axiom should be checked with respect to OWA or CWA, respectively. Thereby, the user has to enter each axiom only once.

Adding a syntactical OWA/CWA axiom annotation leads to two different semantics, which must be clearly separated from each other. To achieve that, we put the closed world interpretation in a special purpose box (see 2.1.4 Special Purpose Boxes). The details are given in Chapter 4. The Approach.

### 3 HYBRID OWA/CWA CHECKS

In this chapter we introduce our approach for hybrid OWA and CWA checks.

From a conceptual point of view, the following three layers TBox, ABox and CBox are relevant for our approach (see Figure 2):

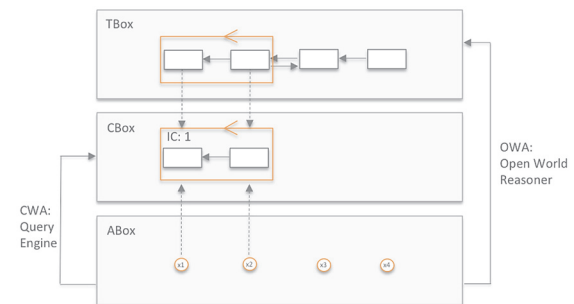


Figure 2: Conceptual View.

- **TBox:** The TBox contains the terminological knowledge in form of general concepts.
- **ABox:** The ABox covers the assertational knowledge in form of individuals which are specified according to the TBox concepts.
- **CBox:** The CBox is a contribution of this paper and covers the constraint knowledge in form of ICs. The ICs are derived from the TBox concepts automatically.

From a technical point of view, an open world reasoner like Pellet performs the OWA checks, and a query engine performs the CWA checks (see Figure 2). The OWA checks are well established in tools

like Protégé (Knublauch and Ferguson, 2004) or TopBraid Composer (Waldenmaier, 2011). Unfortunately, no adequate mechanisms for CWA checks are available to perform the CWA checks automatically. Both Protégé as well as TopBraid Composer support querying over SPARQL queries, but not in an automated way. The users still have to insert the SPARQL query with the appropriate transfer parameter. This remains a maintenance burden, and users need to have knowledge about SPARQL. Thus, in this paper we introduce an approach for applying the CWA checks automatically.

As motivated in Section 2, we follow the approach of Tao et al. and use the DCQ<sup>not</sup> mechanism for the closed world transformation. Specifically, we use the DCQ<sup>not</sup> mechanism (Tao et al. 2010) for defining the closed world semantics exemplarily for the introduced scenario (section 3. Scenario). We have applied the rules for the transformation of a number of most relevant OWL 2 axioms in order to reduce the problem of IC validation to query answering. As query language we use SPARQL (DuCharme 2013), as it is the most widely-used query language on the Semantic Web and enables querying over OWL ontologies via OWL entailment regimes. Furthermore, SPARQL can express DCQ<sup>not</sup>. This is due to the fact that it has the same expressive power as non-recursive Datalog programs (Angles and Gutierrez, 2008).

Suppose we want to make sure that each interface is plugged in to a provider. Then, we have the following SROIQ axiom and we transform the axiom to a DCQ<sup>not</sup> query via applying the mapping rules introduced in (Tao et al., 2010).

$$\begin{aligned} T(\text{Interface} \sqsubseteq \exists \text{isPluggedIn} . \text{Provider}) \\ &:= T_C(\text{Interface}, x) \wedge \text{not } T_C(\exists \text{isPluggedIn} . \text{Provider}, x) \\ &:= \text{Interface}(x) \wedge \text{not}(\text{isPluggedIn}(x, y) \wedge T_C(\text{Provider}, y)) \\ &:= \text{Interface}(x) \wedge \text{not}(\text{isPluggedIn}(x, y) \wedge \text{Provider}(y)) \end{aligned}$$

The result is then transferred to a SPARQL ASK-NOT-EXISTS query. In the following, we explain the definition of the SPARQL templates that we have specified for almost all OWL 2 axioms and that are used in our approach. We have specified ASK-NOT-EXISTS templates, which are based on the DCQ<sup>not</sup> approach from Tao and return “true” in case an individual exists that does not fulfill the constraint. Hence, it returns “false” if all individuals fulfill the constraint.

In this section we have introduced an approach for automated and hybrid OWA and CWA checks for simple OWL 2 axioms.

It is possible to define a complex expression in

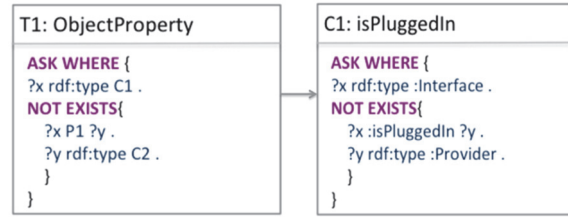


Figure 3: SPARQL template.

form of a SPARQL query manually and use it as IC for a complex expression. However, this requires manual effort. Thus, there is still an open issue with regard to automate complex expressions.

Thus, it needs to be verified, if the approach is scalable with regard to complex formulas or which conceptual modifications need to be done for an automatic transformation. The approach from Tao et al. might be applicable for expressing complex formulas, because currently her prototype reads ICs, which are expressed as OWL axioms and translates each IC first into a DCQ<sup>not</sup> query and then into a SPARQL query.

Further consideration is needed regarding the necessity of complex formulas. It might be the case that engineers do only use simple expressions, especially if they are using customized editors.

Otherwise, it could be possible, that engineers need complex formulas for specific use cases that have to be specified by an expert once and may not modified regularly. Thus, an automated transformation for complex formulas might not be necessarily needed.

## 4 AUTOMATING CWA CHECKS

In this section, we show how the ASK-NOT-EXISTS templates are used by a tool prototype to realize automated CWA checks. Internally, this prototype relies on the following steps:

1. **Load Ontology:** We load the ontology by using the OWL API.
2. **CBox and TBox:** The interplay of the CBox and the TBox is realized with the OWL API. For the SPARQL templates and the automatic transformation from SPARQL to SPIN, we use the SPIN API from TopBraid (Knublauch n.d.). To specify that a selected concept should be interpreted as a closed world axiom (axiom annotation), we use the Jena API (Foundation 2011).
3. **Load ABox:** We use the Jena API for loading the ABox individuals.

**4. Closed World Validation:** For constraint checking we use the Jena API in combination with SPARQL. As mentioned above, our templates are available as SPARQL queries (see Section 4.4 Templates) and are transformed to SPIN templates via the TopBraid API.

In the remainder of this section, we show how a user might work with the tool prototype, again based on the simple connector/plug example.

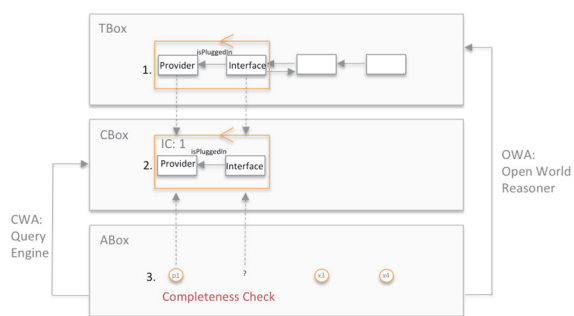


Figure 4: Completeness Check.

- 1. User selects TBox concept:** The user selects the TBox concept ‘Interface isPluggedIn Provider’ (see Figure 4 (1.)) via axiom annotation (Motik et al. 2009) and triggers the closed world transformation.
- 2. System automatically chooses the appropriate template:** The transformation from the TBox concept to the CBox integrity constraint is performed by the system automatically. To choose the appropriate template the system reads out the type of rdf-code of the selected TBox concept. In this case the type of the selected concept is ‘Object Property’ (see Figure 5).

```
<rdf:Description rdf:about="#isPluggedIn">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</rdf:Description>
```

Figure 5: RDF-Code Snippet.

Thus, the SPARQL template ‘Object Property’ is chosen and the transfer parameter of the TBox concept are filled in the template (see Figure 3).

- 3. User triggers Closed World Transformation:** Conceptually, this means that the TBox concepts are interpreted as CBox ICs (see Figure 4 (2.)) and the completeness of the available ABoxes is verified. Thus, the user receives feedback, if all ABoxes are specified correctly, with regard to the chosen CBox IC (see Figure 4 (3.)). In this example the user receives an error message: ‘The ABox instances are incomplete. No ABox instance for the TBox concept Interface exists.

Please complete the ABox accordingly’. Thus, the user adapts the ABox.

This process is performed iteratively until all ABoxes are specified according to the CBox integrity constraint.

## 5 CONCLUSION AND FUTURE WORK

In this paper, we have reviewed the state of the art in OWA/CWA reasoning. Based on this survey, we have selected a number of compatible approaches as the basis of a combination approach for hybrid OWA/CWA checks.

In the future we will transfer this approach to the domain of IT-Security in Critical Infrastructures within the project Networked IT-Security of Critical Infrastructures (Bergner et al. n.d.). Thus, we will specify an ontology for IT-Security in Critical Infrastructures and a methodology for defining threat modelling and measurement handling and we will predefine categories in form of complex templates for modelling threats with regard to derive the appropriate measures automatically.

## ACKNOWLEDGEMENTS

First, I want to thank Dr. Klaus Bergner, Prof. Dr. Andreas Rausch and Dr. Christian Bartelt. It was a pleasure to work with you. A special thanks goes to Axel Mauritz and Carsten Strobel from Airbus Group Innovations in Ottobrunn and to their whole team.

## REFERENCES

Angles, R. and Gutierrez, C., 2008. The expressive power of SPARQL. Available at:  
 Bergner, S., 2015. Methodology for an Ontology-Driven Product Configuration Process. Technische Universität Clausthal.  
 Bergner, S. et al., NETWORKED IT-SECURITY FOR CRITICAL INFRASTRUCTURES – THE RESEARCH AGENDA OF VESIKI Case Studies and IT-Security Matchplays. , pp.1–4.  
 Calvanese, D., Giacomo, G. De and Lembo, D., 2007. EQL-Lite: Effective First-Order Query Processing in Description Logics. IJCAI. Available at:  
 Clark, K. and Sirin, E., 2013. On RDF validation, stardog ICV, and assorted remarks. RDF Validation Workshop. Practical Assurances for ....



- Damáσιο, C. et al., 2006. Supporting open and closed world reasoning on the web. Available at:
- Dix, J. et al., 2003. IMPACTing SHOP: Putting an AI planner into a multi-agent environment. *Annals of Mathematics and ...*
- Donini, F. et al., 1992. Adding Epistemic Operators to Concept Languages. *KR*.
- Donini, F., Nardi, D. and Rosati, R., 2002. Description logics of minimal knowledge and negation as failure. *ACM Transactions on Computational ...*
- DuCharme, B., 2013. *Learning SPARQL* 2nd ed., O'Reilly.
- Etzioni, O., Golden, K. and Weld, D., 1994. Tractable closed world reasoning with updates. *Proceedings of the 4th ...*
- Foundation, T.A.S., 2011. Jena API. Available at: <https://jena.apache.org>.
- Grimm, S. and Motik, B., 2005. Closed World Reasoning in the Semantic Web through Epistemic Operators. *OWLED*.
- Heflin, J. and Muñoz-Avila, H., 2002. Lcw-based agent planning for the semantic web. *Ontologies and the Semantic Web*.
- Hitzler, P. et al., 2007. *Semantic Web: Grundlagen*.
- Hustadt, U., 1994. Do we need the closed world assumption in knowledge representation? *KRDB*.
- Katz, Y. and Parsia, B., 2005. Towards a Nonmonotonic Extension to OWL. *OWLED*.
- Knublauch, H., TopBraid SPIN API.
- Knublauch, H. and Ferguson, R., 2004. The Protégé OWL plugin: An open development environment for semantic web applications. *The Semantic Web- ISWC ...*
- Krishnadh, A., Sengupta, K. and Hitzler, P., 2011. Local closed world semantics: Keep it simple, stupid!
- Lifschitz, V., 1994. Minimal belief and negation as failure. *Artificial Intelligence*.
- Lifschitz, V., 1991. Nonmonotonic Databases and Epistemic Queries. *Proceedings of the 12th International Conference on Artificial Intelligence- Volume 1*, pp.381-386.
- Lloyd, J. and Topor, R., 1984. Making Prolog more expressive. *The Journal of Logic Programming*.
- Lukasiewicz, T., 2004. Combining Answer Set Programming with Description Logics for the Semantic Web.
- Lutz, C., Seylan, I. and Wolter, F., 2013. Ontology-based data access with closed predicates is inherently intractable (sometimes). *Proceedings of the Twenty-Third international ...*
- Mäs, S., Wang, F. and Reinhardt, W., 2005. Using ontologies for integrity constraint definition. *Proceedings of the 4th international ...*
- Motik, B. et al., 2006. Can OWL and logic programming live together happily ever after? *The Semantic Web- Iswc 2006*.
- Motik, B., Horrocks, I. and Sattler, U., 2007. Adding Integrity Constraints to OWL. *OWLED*.
- Motik, B., Horrocks, I. and Sattler, U., 2009. Bridging the gap between OWL and relational databases. ... *Semantics: Science, Services and Agents on ...*
- Motik, B. and Rosati, R., 2006. Closing semantic web ontologies.
- Pan, J. and Ren, Y., 2012. Local closed world reasoning: a personal view on current status and trends. *Journal of Zhejiang University-Science C*.
- Patel-Schneider, P. and Franconi, E., 2012. Ontology constraints in incomplete and complete data. *The Semantic Web- ISWC 2012*.
- Polleres, A., Feier, C. and Harth, A., 2006. Rules with contextually scoped negation.
- Reiter, R., 1992. What should a database know? *The Journal of Logic Programming*.
- Ren, Y., Pan, J. and Zhao, Y., 2010. Closed world reasoning for OWL2 with NBox. *Tsinghua Science and Technology*.
- Schmidt, M. and Lausen, G., 2013. Pleasantly consuming linked data with rdf data descriptions. *arXiv:1307.3419*.
- Seylan, I., Franconi, E. and Bruijn, J. De, 2009. Effective query rewriting with ontologies over DBBoxes. *IJCAI*.
- Sirin, E., Parsia, B. and Grau, B., 2007. Pellet: A practical owl-dl reasoner. *Web Semantics: science, ...*
- Sirin, E., Smith, M. and Wallace, E., 2008. Opening , Closing Worlds - On Integrity Constraints. *Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions*, 432.
- Tao, J., 2010. Adding integrity constraints to the semantic web for instance data evaluation. *The Semantic Web- ISWC 2010*.
- Tao, J. et al., 2010. Integrity Constraints in OWL. *AAAI*.
- Waldenmaier, C., 2011. *TopBraid Composer - Getting Started Guide*.